Development of Dependable Controllers in the Context of Machines Design

J. Machado[1] and J. C. Campos[2]

[1] Minho University, Mech. Eng. Dept, CT2M, Portugal, e-mail: jmachado@dem.uminho.pt

[2] Minho University, Informatics Dept and HASLab, INESC TEC, Portugal, e-mail: jose.campos@di.uminho.pt

Abstract In the domain of machines' design, one of the most important issues to solve is related with the controller's design, mainly, guaranteeing that the machine will behave as expected. In order to achieve a dependable controller, some steps can be considered, such as the formalization of its specification - before being translated to the program that will be inserted in the controller device – and the respective analysis and verification. Nowadays, some formal analysis techniques, such as formal verification, are used to achieve this purpose. The dependability of a controller, however, is impacted by its execution context. This paper proposes an approach for the formal verification of the specification of mechatronic system's controllers, which considers, on the formal verification tasks, the behavior of the plant and the behavior of the Human Machine Interface of the Mechatronic system. Some conclusions are extrapolated for other systems of the same kind.

Key words: Dependable Controllers, Machines' Design, Formal Verification, Human Machine Interface, Partial Plant Models.

1 Introduction

The development of controllers' software for mechatronic systems, when performing all machines' design tasks, is a very complex and exigent process.

A mechatronic system is composed, mainly, by three parts: Controller, Plant and Human Machine Interface (HMI) (see Figure 1). These parts interact and behave together, and the development of the software to be introduced in the controller, must take into account the behavior of those parts and the interrelation between them.

Several steps can be performed in order to obtain a dependable controller: first, the use of methodologies for obtaining the structure of the controller's specification [1]; second, the use of a formalism to describe, formally, the intended behavior for the controller [2]; third, the use of analysis techniques, in order to guarantee the dependability of the specification [3]; and, fourth, the translation of the specification into a controller program and respective implementation on a physical controller [4].

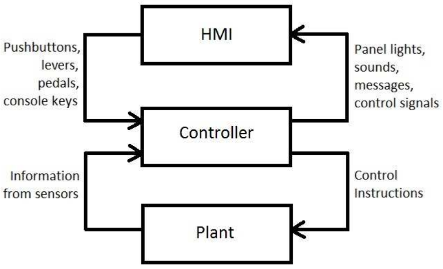

Fig. 1. General configuration of a mechatronic system.

The first two steps are very well studied and there are formalisms and tools that can be used to perform them. For the structure of the controller specification it is possible to use GEMMA [5] or Multi-Agent formalisms [6], for example. For the specification formalisms such as Petri Nets [7], SFC [8] or Statecharts [9] can be used. For the implementation, PLCs [10] and Microprocessors [11], among others.

If we consider step 4, as being systematic, the most important step remains step 3, where the specification must be tested, simulated and verified.

This paper proposes a methodology to obtaining a dependable specification, which is focused on the analysis of the specification by using formal verification techniques. The paper discusses the relevance of considering aspects related with the behavior of the plant, and the behavior of the HMI, when developing the specification for the controller of a mechatronic system.

The paper is organized as follows: section 2 provides an overview of the main steps considered in the design of a mechatronic system; section 3 proposes a case study to be used in the ensuing discussion; section 4 discusses aspects related to considering plant behavior in the development of a specification; further, section 5 discusses aspects related to considering the HMI behavior in the development of a specification; and finally, section 6 presents some conclusions and future work.

2 Steps for designing a mechatronic system

The development of controllers' software for mechatronic systems, raises a number of challenges. From the desired behavior specifications, until the implementation of a controller program for a mechatronic system, the controller designer needs to use some different and complementary formalisms and tools that help him in all the necessary steps. Taking into account aspects related to systems' dependability, the designer must be able to use together these formalisms and tools in order to achieve the desired behavior for the system.

From the analysis of needs, passing by the conception, realization into the implementation and exploitation of a mechatronic system, there are several steps that must be realized (Figure 2). During each step of the controller's design, a corresponding step exists that relates to the development of the plant (physical part of the system: motors, cylinders, sensors, …). For instance, step 3 corresponds to the specification of the controller and the step 3' corresponds to the specification of the plant.

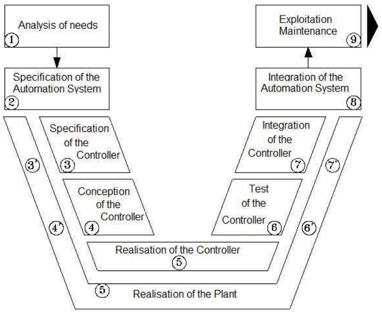
Analysis of needs
1
Specification of the Automation System
2
Specification of the Controller
3
3'
Conception of the Controller
4
4'
Realisation of the Controller
5
5
Realisation of the Plant
Test of the Controller
6
6'
Integration of the Controller
7
7'
Integration of the Automation System
8
Exploitation Maintenance
9

Fig. 2. Steps considered on the design of a mechatronic system.

The approach presented in this paper is focused on the steps 3, 4 and 5 of the Figure 2.

3 Case study

The chosen system for this case study lies in the well-known category of "pick-and-place" systems (Figure 3).

Its function is to take parts, fed by gravity into three feed chutes, for placement in a single unloading chute. Sensors pp1, pp2 and pp3 indicate the presence of a part in one of the feed chutes, while sensor pp0 signals the presence of a part in the unloading chute. The device that enables picking and placing a part is composed of a group of three pneumatic cylinders plus a vacuum suction cup system. The vertical cylinder (VC) places the suction cup in contact with a part. Longitudinal cylinders L1C and L2C are arranged in series to allow positioning the vertical cylinder VC in front of the four chutes (L2C stroke is twice as long than L1C stroke). The four reached positions are thereby detected by position sensors s0, s1, s2 and s3. The depression in the suction cup is obtained by virtue of a venturi and detected by a vacuum sensor.

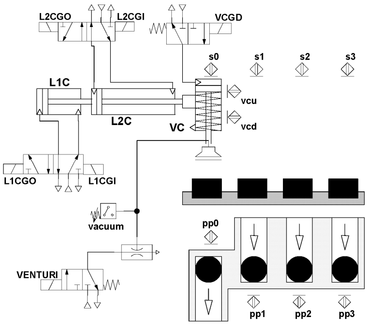

Fig. 3. Pick and place system.

4 Specification analysis considering plant behavior

As part of a dependable controller design approach, the system being targeted for verification can thus be [12] either the controller on its own, presumed to be operating within an open loop on the plant (a non "model-based" verification), or the {controller + plant} assembly set interacting within a closed loop ("model-based" verification).

One problem with model-checking is related to the state explosion problem. The state of the model may become too big for verification to be feasible with reasonable resources. In this paper we report on results of work on mode-based verification resorting to partial models of the Plant. This enables the use of smaller models, thus making it possible to verify larger systems.

This solution allows us, also, a stronger proof of safety properties, which will become stronger as much as the plant model is reduced [13].

Considering the approach proposed in [13], formal verification tasks can be performed with the assumption of a closed loop behavior of the controller model and the plant model. Also, in the same work, it is proposed that a possible solution for obtaining the plant model for this system is considering a set of plant modules, and the solution proposed is the combination of twelve plant modules in order to obtain a modular solution for the entire system plant model.

In [13] a set of behavior properties for the exposed system is considered, to be proven using verification by model-checking. This set of properties is composed by *safety properties* and *liveness properties*. The same work proposes a systematic approach to prove the set of properties using, or not, the plant model of the system, depending on the specific type of property under consideration. It was observed that some safety properties were not proved without a plant model, but were proved when the entire plant model was used.

Hence, in this paper, we analyze what happens if only a part of the plant model is used. For instance, consider the following behavior property: “*While the vertical cylinder is moving down, all the other cylinders stay in deployed or retracted position*”. This property cannot be proved without a model of the plant. With the full model of the plant, the property took 109 minutes to prove [13], using the NuSMV model-checker, and a machine with a Pentium III processor at 1 GHz and 1 GB of RAM.

When we intend to prove the property, the use of the entire plant model is not a good solution, for two reasons: first, because the proof of safety properties will become stronger as the plant model is reduced [13] and, second, because the global model becomes bigger and more difficult to analyze by the model-checker.

Considering this property, which deals only with the three cylinders of the system, it seems enough to use only the models of these three cylinders. Using the same machine for calculations, the same Model-checker NuSMV and, now, considering a partial plant model - composed by the models of the three cylinders of the system - the property can be proved only in 12 minutes (about 10% of the time needed if the entire plant model is considered).

Indeed, this smaller plant module is enough to prove the property. However, this cannot be adopted as a systematic rule. Ongoing work is showing us that, for some properties, this rule cannot be applied.

5 Specification analysis considering HMI behavior

Above, the role of a plant model in the verification of a controller’s specification was discussed. The plant, however, is not the only factor affecting the behavior of the controller. When a human operator is present, the operator’s actions at the HMI are also relevant. See for example the discussion in [14].

Consider the example discussed in the previous sections. If a HMI control panel is present that allows the operator to start/stop the system at any time, then verification of a property like: “*a picked part will always be placed in the unloading chute*” will always fail because the operator might stop the system (see the discussion in [15]). In order to avoid this, a model of the operator needs to be included. If a strict operations procedure is assumed, this model is easy to express. It describes the sequences of operations as prescribed by the operations procedure. This type of model, however, is usually too restrictive.

Typically problems will arise when the operator deviates from the prescribed procedure. In the more general context of interactive computing systems this is addressed by describing the possible operator responses to the different output of the system [16]. Hence it could be expressed that the operator will only stop the system if no part is currently picked up. Under that assumption, the proof becomes feasible. In this approach, proving the safety of the system implies deriving and making explicit assumptions about how the operator must behave. In doing so, we identify potential points of failure and areas where the controller might need to be improved to account for user error. Hence, we could improve the controller to only act on a stop signal from the operator after the currently held part is placed in the unloading chute.

6 Conclusions

This paper illustrated how consideration of partial plant models (instead of a global plant model) and of operator models can be useful for the formal verification of Industrial Mechatronic Systems' Controllers.

Even if it is possible to see the importance of this approach, we have not developed, yet, a systematic approach to finding out, quickly, which models must be considered in order to verify a specific behavior property of the system. This will be the next step of this project.

References

1. Machado, J. M.; Seabra, E.: A Systematized Approach to Obtain Dependable Controllers Specifications, *Proc. of 20th International Congress of Mechanical Engineering*, 2009.
2. David, R.: Grafcet: a powerful tool for specification of logic controllers. *IEEE Transactions on Control Systems Technology*, 3, 1995, pp. 253-268.
3. Johnson, T. L.: Improving automation software dependability: A role for formal methods? *Control engineering practice*, 15(11), 2007, pp. 1403-1415.
4. Machado, J., Denis, B., Lesage, J.-J., Faure., J.-M. and Silva, J. F. D.: Logic controllers dependability verification using a plant model. *Proc. of 3rd IFAC Workshop on Discrete-Event System Design (DESDes)*, September 2006, pp. 37-42.
5. ADEPA: GEMMA – ADEPA, France, 1992.
6. Sohier, C.: Pilotages des Cellules Adaptatives de Production: Apport des Systemes Multi-Agents. PhD Thesis, École Normale Supérieure de Cachan, Paris, France, 1996.
7. Murata, T.: Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, 77 (4), April 1989, pp. 541-580.
8. IEC: IEC 60848 - GRAFCET Specification Language for Sequential Function Charts. Edition 2.0 b, 2002.
9. Harel D., 1987, "Statecharts : a visual formalism for complex systems", Science of computer programming North Holland, Vol. 8 pp 231-274.
10. Moon, I.: Moddeling Programmable Logic Controllers for Logic Verification, *IEEE Control Systems Magazine*, 1994, pp 53-59.
11. Brusamolino, M., Reina, L. and Spalla, M.F.: An example of microprocessor's application in minicomputer systems: a copy volume design and implementation, *Microprocessing and Microprogramming*, 13 (5), May 1984, pp. 331-339.
12. Frey, G. and Litz, L.: Formal methods in PLC programming. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics (SMC 2000*), 2000.
13. Machado, J., Denis, B. and Lesage, J.-J.: A generic approach to build plant models for DES verification purposes. *Proceedings of the 8th International Workshop On Discrete Event Systems (WODES'06)*, July 2006, pp. 407-412.
14. Leveson, N. G.: Safeware: System Safety and Computers. Addison-Wesley, 1995.
15. Campos, J. C. and Harrison, M.D.: Model Checking Interactor Specifications. *Automated Software Engineering*, 8(3), 2001, pp. 275-310.
16. Doherty, G. J., Campos, J. C. and Harrison, M. D.: Resources for Situated Action. *Lecture Notes in Computer Science*, 5136, Springer, 2008, pp. 194-207.